

APPLICATION FOR UNITED STATES PATENT

**METHODS AND SYSTEMS FOR MANAGING
COMPLEX PROJECTS OVER A COMPUTER
NETWORK THROUGH COLLABORATIVE
DEPENDENCY MANAGEMENT**

Inventor: Kurt G. ROBSON
551 Trinidad Lane
Foster City, CA 94404
Citizenship: US

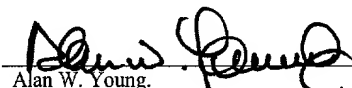
Inventor: Girish R. DIVATE
32417 Pacific Grove Way
Union City, CA 94587
Citizenship: India

Inventor: Cedric M. NG
15 Mulberry Ct. #16
Belmont, CA 94002
Citizenship: US

Assignee: ORACLE CORPORATION
500 Oracle Parkway,
Redwood Shores CA 94065

Young Law Firm, P.C.
4370 Alpine Road, Suite 106
Portola Valley, CA 94028
Tel: (650) 851-7210
Fax: (650) 851-7232

I hereby certify that this patent application is being deposited with the United States Postal Service as Express Mail No EL 833 860 097 US in an envelope addressed to Assistant Commissioner for Patents, Box Patent Application, US Patent and Trademark Office, Washington, D.C., 20231, on May 29, 2001.

 May 29, 2001
Alan W. Young. Date

METHODS AND SYSTEMS FOR MANAGING COMPLEX PROJECTS OVER A COMPUTER NETWORK THROUGH COLLABORATIVE DEPENDENCY MANAGEMENT

5

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to the management of complex projects. More particularly, the present invention relates to the field of project schedule management for large and complex projects and to methods and systems designed to enable a collaborative and distributed maintenance of project schedules.

2. Description of the Related Art

The development of a new product, whether tangible or intangible (such as a new service) is often mapped out in a project, which sets out an anticipated timeline for taking the product or service from concept through final implementation, distribution or production. Such projects may be phenomenally complex at their inception or may become so during the lifetime thereof, particularly in the high technology, aerospace, defense and financial services industries, to identify but a few examples. Software tools exist that help project managers define the project and an anticipated project timeline. However, such tools typically give the manager only a macro-level view of the project and of the constituent tasks thereof. More often than not, however, projects do not docilely follow their anticipate timeline, as set out at the beginning of the project. Indeed, unanticipated problems arise in all but the simplest projects. As such problems arise and are addressed, the originally defined project timeline and the actual progress of the project over time diverge and may ultimately bear little useful relation to one another. Moreover, as the complexity of the project rises, the burden of

updating the project schedule may become a significant drain on resources, further eroding its perceived usefulness in the eyes of those tasked with managing the project. When this happens, critical updates may not be incorporated into the project schedule, which further decreases its utility as a predictor of project completion and as a project status indicator.

5 Large and complex projects may involve hundreds or thousands of people, and are often widely distributed, not only across geographical and political boundaries, but also across enterprise boundaries and time zones. More often than not, the contributors to a project do not have (and need not have) the same level of access to the project details and schedule. Compartmentalization is commonly used to segregate project contributors for a
10 variety of reasons, such as to insure security. The consequence of such compartmentalization, however, is that project contributors do not have the access required to determine the relative importance of the task assigned to them within the project. As most tasks within a project are connected to many others, a failure or delay in even a seemingly low-level task may have profound repercussions in higher level tasks as the effect of that
15 failure or delay ripples up the project hierarchy.

What are needed, therefore, are improved project scheduling tools that enable project contributors to dynamically update the project definition and timeline. What is also needed are methods and systems to enable potentially widely disseminated project contributors to update the status of their assigned task in a secure manner and in a manner that insures that
20 the overall project timeline accurately describes the current status of the entire project and its constituent tasks and functions as an accurate predictor of the project completion. Also needed is a scheduling tool for large and complex projects that is sufficiently flexible as to enable project contributors to define new tasks and problem Issues and that enables the

tracking of the progress of such new task and Issues, from definition through resolution thereof.

SUMMARY OF THE INVENTION

It is, therefore, an object of the present invention to provide improved project
5 scheduling tools that enable project contributors to dynamically update the project definition
and timeline. It is another object of the present invention to provide methods and systems to
enable potentially widely disseminated project contributors to update the status of their
assigned task in a secure manner and in a manner that insures that the overall project timeline
accurately describes the current status of the entire project and its constituent tasks and
10 functions as an accurate predictor of the project completion. A still further object of the
present invention is to provide a scheduling tool for large and complex projects that is
sufficiently flexible as to enable project contributors to define new tasks and problem Issues
and that enables tracking the progress of such new task and Issues, from definition through
resolution thereof.

15 In accordance with the above-described objects and those that will be mentioned and
will become apparent below, a method of managing a project that includes a plurality of
interdependent tasks according to the present invention, may include steps of defining a first
and a second task and storing the first and second tasks in a database stored in a server that is
selectively and remotely accessible over a computer network, each of the first and second
20 tasks having a status associated therewith; defining a first dependency relationship between
the first and the second tasks and storing the defined first dependency in the database;
enabling remote updating of the status of the first and/or second tasks over the computer
network; enabling remote definition of an Issue, a Change Request and/or a Change Order,

the Issue identifying a problem within the project whose resolution is to be tracked, the Change Request identifying one or more steps to be taken pending authorization to resolve Issue and the Change Order identifying authorized steps to resolve the Issue; storing the defined Issue, Change Request and/or Change Order in the database and enabling remote definition of one or more second dependency relationships between the defined Issue, Change Request and/or Change Order and the first and/or second tasks and storing the defined second dependency in the database.

According to further embodiments, a step of defining a permission that defines access rights for the first task, the second task, the Issue, the Change Request and/or the Change Order may be carried out. The permission further may define a right to remotely change the status of the first task and/or the second task and/or to remotely change the first dependency relationship between the first and second task. The first and/or second dependency relationships may be selected from a group including Start-Start, Start-Finish, Finish-Start and Finish-Finish, for example. The first and/or second dependency relationships may include a definition of a lag time between the start and/or finish of at least two of the defined first and second tasks, the Issue, the Change Request and the Change Order. The problem identified by the Issue may be a problem that was previously unidentified at the time when the first and second tasks were defined. Each of the defined Issue, Change Order and Change Request may include a status associated therewith and the method may further include a step of enabling remote updating of the status of the defined Issue, Change Request and/or Change Order. The status of the first task, the second task, the defined Issue, Change Order and Change Request may be selected from a group including: not started, on track, complete, in trouble, on hold and cancelled, for example. The permission may further define a right to

remotely change the stored second dependency relationship(s). A step may be carried out to maintain a selectively and remotely accessible graphical representation of the first task, the second task, the first dependency relationship, the defined Issue, Change Request and/or the Change Order and/or the second dependency relationship, among other items. The

5 selectively and remotely accessible graphical representation may be rendered on a Web browser or other suitable interface. A step may be carried out to define and store in the database an identity of at least one entity allowed access to and/or having responsibility for each of the defined first and second tasks, the defined Issue, Change Request and/or the Change Order. The entity, for example, may be selected from a group including a project

10 team, a project member, a subcontractor and a vendor. The computer network may include the Internet, for example, or may be a private computer network. The computer network's reach may cross time zone, geographical and/or enterprise (corporate) boundaries. The graphical representation may include, for example, a selectively expandable hierarchical tree (with an unlimited number of levels) that shows the first task, the second task, the first

15 dependency relationship, the defined Issue, Change Request and the Change Order and/or the second dependency relationship. A step may be carried out to prompt the user for the remote definition of an Issue, Change Request, Change Order and/or the second dependency relationship(s) when (for example) the status of a task is updated. Steps may be carried out to enable a document to be associated with the defined first and second tasks, the Issue, the

20 Change Request and/or the Change Order and to enable the associated document to be stored in the database.

According to another embodiment thereof, the present invention is also a method of participating in a project that includes a plurality of interdependent tasks, comprising the

steps of accessing a remote database over a computer network, the database storing: a definition of a first and a second task, a status associated with each of the first and second tasks, and a first dependency relationship between the first and the second task; updating of the status of at least one of the first and second tasks over the computer network; defining
5 and causing to be stored in the database at least one of an Issue, a Change Request and a Change Order, the Issue identifying a problem within the project whose resolution is to be tracked, the Change Request identifying at least one step to be taken pending authorization to resolve Issue and the Change Order identifying authorized steps to resolve the Issue; defining one or more second dependency relationships between the defined Issue, Change Request or
10 Change Order and the first and/or second tasks and causing the defined second dependency to be stored in the database.

The present invention may also be viewed as a computer system configured for managing a project timeline that includes a plurality of interdependent tasks, comprising: at least one processor; at least one data storage device; a plurality of processes spawned by the
15 at least one processor, the processes including processing logic for: defining a first and a second task of the plurality of tasks and storing the first and second tasks in a database stored in a server that is selectively and remotely accessible over a computer network, each of the first and second tasks having a status associated therewith; defining a first dependency relationship between the first and the second tasks and storing the defined first dependency in
20 the database; enabling remote updating of the status of the first and/or second tasks over the computer network; enabling remote definition of an Issue, a Change Request and/or a Change Order, the Issue identifying a problem within the project whose resolution is to be tracked, the Change Request identifying at least one step to be taken pending authorization to resolve

Issue and the Change Order identifying authorized steps to resolve the Issue; storing the defined Issue, Change Request and/or Change Order in the database; enabling remote definition of one or more second dependency relationships between the defined Issue, Change Request or Change Order and the first and/or second tasks and storing the defined second
5 dependency in the database.

The present invention, according to another embodiment thereof, is a machine-readable medium having data stored thereon representing sequences of instructions which, when executed by computing device, causes the computing device to manage a project timeline that includes a plurality of interdependent tasks by performing the steps of: defining
10 a first and a second task and storing the first and second tasks in a database stored in a server that is selectively and remotely accessible over a computer network, each of the first and second tasks having a status associated therewith; defining a first dependency relationship between the first and the second tasks and storing the defined first dependency in the database; enabling remote updating of the status of the first and/or second tasks over the
15 computer network; enabling remote definition of an Issue, a Change Request and/or a Change Order, the Issue identifying a problem within the project whose resolution is to be tracked, the Change Request identifying at least one step to be taken pending authorization to resolve Issue and the Change Order identifying authorized steps to resolve the Issue; storing the defined one of the Issue, Change Request and Change Order in the database; and enabling
20 remote definition of at least one second dependency relationship between the defined Issue, Change Request or Change Order and the first and/or second tasks and storing the defined second dependency in the database.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates aspects of a system and method for managing a complex project, according to an embodiment of the present invention.

Fig. 2 shows a project task hierarchy, according to an embodiment of the present invention.

Fig. 3 shows an example of a Web browser displaying illustrative aspects of the present invention.

Fig. 4 is a flowchart of a method for managing complex projects, according to an aspect of the present invention.

Fig. 5 is a flowchart of a method for managing complex projects, according to another aspect of the present invention.

Fig. 6 is a diagram of a computing device with which the present invention may be practiced.

DESCRIPTION OF THE INVENTION

FUNCTIONAL DESCRIPTION

Fig. 2 shows a project task hierarchy 200, according to an embodiment of the present invention. The constituent tasks of the project hierarchy 200 may be logically organized and represented as a hierarchical tree structure. The tree structure, as shown in Fig. 2, may include a root task, such as shown at Task₀. Task₀, for example, may represent the ultimate goal of the project, such as the manufacture of airliner, the coding of a large software product, the design of an automobile or some other large and complex project. As shown, Task₀ has a number of predecessor tasks that must be completed before Task₀ may be considered to have been completed. The predecessor-successor relationship between the

constituent tasks of the project 200 are indicated by the arrow 218 labeled "Time". Thus, Task₁₁, Task₁₂ and Task₁₃ (and possibly others, as suggested by the dashed lines) are predecessor tasks to Task₀. In turn, each of the Task₁₁, Task₁₂ and Task₁₃ may have a number of predecessor tasks. For simplicity of illustration, only the predecessor tasks of Task₁₁ are shown in Fig. 2. These include the tasks labeled Task₂₁, Task₂₂ and Task₂₃. Likewise, completion of Task₂₂ may be related to predecessor tasks Task₃₁, Task₃₂ and Task₃₃. It is understood that the hierarchical tree structure 200 is but a partial illustration of an example of a large project. In practice, projects may be considerably more complex than suggested by Fig. 2 and the present invention is drawn to managing such complex projects, using the systems and methodologies detailed herein.

According to the present invention, each of the tasks shown in Fig. 2; namely Task₀ to Task₃₃, is related to at least one other task within the project by a selectable dependency relationship. For example, the dependency relationships may be categorized by the following non-exclusive and exemplary list: Start-Start (SS), Start-Finish (SF), Finish-Start (FS) and Finish-Finish (FF). These dependency relationships may be understood by assigning the first letter of the combinations (SS), (SF), (FS) and (FF) to the "Started or Finished" status of the current task and the second letter the "Started or Finished" status of the successor task. Should the dependency relationship between two tasks be selected to be SF, for example, the current task may not be started until its successor task is finished. Likewise, a dependency relationship of FS indicates that the current task must be finished before its successor task may be started. Similarly, SS, indicates that the current task must be started before work may begin on its successor task (both tasks should, therefore, run concurrently). Lastly, a dependency relationship of FF between tasks means that both the current task must be

finished before its successor task is finished. Other dependency relationships may be defined and implemented within the context of the present invention, as those of skill in this art may recognize. Alternatively, the above-detailed dependency relationships (SS, SF, FS and FF) may refer to the current task and the current task's predecessor task. Whichever convention is
5 chosen, however, should preferably be adopted for all of the dependencies throughout the project.

Large projects, by their very nature, may not be fully definable at the project inception. That is, each constituent task of the project may not be defined at the start of the project. Problems can and frequently do arise in complex projects, and these problems,
10 whether on the project critical path or not, may be interrelated to other tasks within the project. Moreover, the impact of delays caused by such unanticipated problems may not be immediately apparent to the person or group encountering the problem or, more egregiously, to the project management. Such problems, unless tracked within the context of the other defined tasks of the project, have the capacity to derail the entire project timeline or key
15 portions thereof. One of the major responsibilities of project managers is to accelerate the priority of selected tasks, as it is often only the project manager (or the managers of specific portions of the project) that is privy to the macro-level view of the project necessary to identify potential problem areas and to take the requisite preemptive measures. If unanticipated problems arise and are not integrated within the larger project management
20 framework, critical dates may slip and the timeliness of completion of the project may be in jeopardy.

The present invention, therefore, provides methods and systems to track and manage such unanticipated problems by integrating them into the project schedule, as set forth below.

Referring back to Fig. 2, the present invention provides the ability to define an Issue and to integrate that Issue within the project management framework. Graphically, such an Issue (shown in Fig. 2 as Issue₂₁₂) may be illustrated as being inserted within the hierarchical tree structure 200, thereby enabling the Issue to be assigned appropriate relationship dependencies to the existing tasks (and/or other Issues) within the tree structure 200. In the illustrative example developed in Fig. 2, the present invention provides for the ability to insert Issue₂₁₂ (representing a previously unidentified problem or unresolved Issue) into the hierarchical tree structure 200 between, for example, Task₂₂ and the tasks Task₃₁, Task₃₂ and Task₃₃. Moreover, the present invention also provides for defining the dependency relationship of the inserted Issue with its predecessor and successor task(s) and/or other Issue(s) within the hierarchical tree structure 200 or other project management framework. This ability to insert an Issue into the task hierarchy not only enables project managers to manage the newly discovered issue, but also directly involves the person(s) responsible for resolving the Issue more fully in the project by requiring that the dependency relationships between the new Issue and pre-existing tasks and/or other Issues be identified and integrated into the project hierarchy 200.

According to the present invention, resolution of an Issue such as Issue₂₃₂, may involve nothing more than a delay in providing some deliverable item, for example. In that case, when the item is delivered, the Issue may be fully resolved, or the delay may trigger cascading delays in other tasks and/or Issues. However, to resolve an Issue, the execution of specific steps may be required. Such steps may be carried out without further input from the project management. Alternatively, the steps required to resolve the Issue may be such as to require some level of authorization from some level of the project management team. In such

a case, the Issue may evolve into (or may be modified to include) a Change Request that outlines one or more proposed steps to resolve the previously identified Issue, as shown at 214 in Fig. 2. Such a Change Request may include a document associated with the identified Issue and listing proposed steps to resolve the Issue. When and if authorization is obtained to implement the changes outlined in the Change Request or to execute the proposed steps listed therein (or modified versions thereof), the Change Request 214 may evolve into (or be replaced by) a Change Order, as shown at 216. A Change Order, according to the present invention, identifies the changes or steps that have been authorized by the relevant authority to resolve the Issue, such as Issue₂₁₂. As shown in Fig. 2, a dependency relationship 212 may be defined between the newly defined Issue₂₁₂, the Change Request 214 and/or the Change Request 216 and its successor and/or predecessor task(s). In the example shown in Fig. 2, reference 212 is identified as being a dependency relationship of the FS type, meaning that Issue₂₁₂ must be completed (i.e., resolved) before Task₂₂ may be started. Therefore, resolution of the newly discovered Issue₂₁₂ may be critical to the project, as Task₂₂ may not even be started until Issue₂₁₂, Change Request 214 and/or Change Order 216 is resolved/completed. Moreover, resolution of Issue₂₁₂ may itself require the definition and resolution of one or more additional Issues and/or Change Requests and/or Change Orders. Therefore, the person or entity assigned to resolve any Issue, Change Request and/or Change Order may, according to the present invention, define and integrate into the project hierarchy other Issues, Tasks, Change Requests and/or Change Orders upon which the resolution and completion the resolution and completion of their assigned Issue, Change Request and/or Change Order depends. This ability to define and integrate additional (and a priori unforeseen) Tasks, Issues, Change requests and/or Change Orders into the multi-level

hierarchical structure 200 provides great functionality and flexibility. Indeed, the hierarchy 200 may be expanded as needed at any time during the project execution to a far more granular level than the initially defined hierarchy at the project inception.

For example, as shown in Fig. 4, Task₂₂ may have been originally defined and assigned to a person and/or organization in the initial project plan. Thereafter, as discussed above, it may be discovered that completion of Task₂₂ requires resolution of Issue₂₁₂, a heretofore-unknown problem or issue. Issue₂₁₂, therefore, may be defined, assigned to a person or organization and integrated into the hierarchy 200. In turn, those assigned to complete Issue₂₂₁, may define Task₂₂₁ as the or one of a plurality of Tasks whose completion is necessary to resolve Issue₂₁₂. Task₂₂₁ may, therefore, also be integrated into the hierarchy 200. Upon tackling the resolution of Task₂₂₁, those responsible for Task₂₂₁ may discover, define, integrate and assign other Tasks and/or Issues, such as suggested by Task₂₂₁₁, Issue₂₂₁₂ and Issue₂₂₁₃. Therefore, the resolution and completion of Task₂₂₁₁, Issue₂₂₁₂ and Issue₂₂₁₃ may be selectively visible and tracked by those who have responsibility for hierarchically higher issues and tasks, as well by project managers having the ultimate responsibility for the project or portions thereof. In turn, during execution of Task₂₂₁₁, a new Issue may be identified and integrated, as shown at Issue₂₂₁₁₂. As shown, all of the aforementioned hierarchically lower Tasks and/or Issues (and any corresponding change Requests or Change orders) must be completed and/or resolved for the Task₂₂ to be completed. It is quite likely, moreover, that most (if not all) of these hierarchically lower Issues and/or Tasks were unknown (or not clearly defined or definable) at the time the initial project plan was developed. The present invention, therefore, dynamically supports any level of Tasks, Issues, Change Requests and Change Orders. That is, the hierarchical structure 200

according to the present invention may be caused to dynamically expand as needed, contemporaneously with the identification of previously unknown Tasks and/or Issues.

Each of the newly defined and integrated Tasks, Issues, Change Requests and/or Change Orders may be assigned to a specific person or entity who may be given primary responsibility for the resolution and completion of the newly defined and integrated Task, Issue, Change Request and/or Change Order. The present invention may also advantageously be configured to send a message (such as by email, for example) to the person assigned to any given Task, Issue, Change Request and/or Change Order. The message may be automatically sent via a workflow and Web-based system before the due date of the Task, Issue, Change Request and/or Change Order to remind and/or prompt for changes in the status and estimated completion dates thereof. Automated email-based messaging is highly useful when the resolution of one or more Tasks, Issues, Change requests and/or Change Orders depends upon actions of people or organizations that are widely scattered across multiple organizations, countries and/or time zones.

Fig. 3 shows an example of a Web browser displaying illustrative aspects of the present invention. According to the present invention, the present method of managing a complex project may advantageously be implemented, rendered on and operated using a standard Web browser 300, such as available from Microsoft or Netscape corporations, for example. As shown, the browser 300 may display, in its active window, a number of tasks, Issues, Change Requests and/or Change Orders in tabular form, as shown at 310. Such tabular representation is but one possible graphical representation for visualizing the constituent tasks, Issues, Change Requests and Change Orders of the project. Another representation is the Gantt chart, an exemplar of which is also shown at reference numeral

320 in the active window of the browser 300. Both the tabular representation 310 and the Gantt 320 chart convey similar information, albeit in different forms. Indeed, each task (any of the generically illustrated tasks in Fig. 3 may include an Issue, a Change Order or a Change Request according to the present invention) is uniquely identified by a task name, a start date, an identification of successor task(s) (if any), the type of dependency relationship (SS, SF, FS, FF, for example) between the task and its successor task(s), the task's finish date, an identification of the task's predecessor task(s) (if any) and the type of dependency relationship (SS, SF, FS, FF, for example) between the task and its predecessor task(s).

The present invention takes advantage of the growing ubiquity of the Internet and Web browsers to provide a method that enables a decentralized and collaborative management and maintenance of project schedules. Indeed, large projects may involve more than one organization and may be widely distributed over across time zones, geographical and enterprise boundaries. Using a combination of a database that is remotely accessible over a computer network and Web browsers enables the present invention to implement a server and thin client relationship between the database and the users thereof accessing the database to update the status of tasks and/or to define new Issues, Change Requests and/or Change Orders.

Fig. 1 shows a representation of the method and system 100 for managing complex projects that include a plurality of interdependent tasks, according to an embodiment of the present invention. As shown therein, the system 100 includes a computer network 112, a database 110 that is remotely accessible via a server 111 coupled to the network 112 and one or more computers 118, personal Digital Assistants (PDAs) 119, Web-enabled mobile telephones 120 and/or other so-called thin clients 121 capable of accessing the network 112

and of running some version of a Web browser software 116 or some other suitable interface.

According to the present invention, all of the participants in the project are loosely coupled to the database 110 through the computer network 112. The computer network 112 may include the Internet, for example, and/or other communication infrastructure. Indeed, the terms

Internet and computer network are used interchangeably herein, and are intended to cover the Internet as it exists as of the time of this writing, and also any other communication infrastructure (public or private) for computers and/or other network devices yet to be developed. Security may be assured by suitable encryption of all communications and/or by

creating secure Virtual Private Networks (for example) or by other means known to those of skill in this art, such as a Secure Socket layer (SSL). As shown in Fig. 1, each participant in the project throughout the world 114 may update the status of existing tasks of the project, input new tasks within the hierarchical tree structure and/or input new Issues, Change Orders and/or Change Requests through an Internet project management application embodying the present invention, via an Internet browser software 116 or via another suitable interface.

Therefore, with the aid of a personal computer 118, network computer, Internet-enabled personal digital assistant 119, Web enabled mobile telephone 120 or any other device any device 121 equipped with a modem or other network access device allowing remote access to the corporate network or other Internet-enabled appliance, all participants in the project (and any other properly authorized personnel) may asynchronously input and collaboratively update the status of tasks, Issues, Change Requests and/or Change Orders (and the dependencies therebetween) into the project hierarchy by using a suitable browser and pointing the browser to a selected and preferably secure site (a World Wide Web site, for example) established for that purpose. Upon being properly authenticated (through inputting

a username - password combination or other biometrical identification, for example), the project participant may be prompted or given the opportunity to enter and/or to update the above-listed information, as shown at 124.

According to the present invention, the database 110 may store the tasks, Issues, Change Requests and Change Orders for a single project or for multiple projects. The user may retrieve a specific project by entering the necessary information in a project search screen, a simplified example of which is shown at 122 in Fig. 1. Thereafter, the database 110 may be searched and user selectable views of the desired project may be displayed on the user's browser, as shown at 123.

Fig. 4 is a flowchart of a method for managing project that includes a plurality of interdependent tasks, according to an aspect of the present invention. As shown, the method includes a first step S41 in which (at least) a first and second task of the plurality of tasks are defined. Step S42 calls for storing the first and second tasks in a database (such as shown at 110 in Fig. 1) stored in a server (such as shown at 111 in Fig. 1) that is selectively, securely and remotely accessible over a computer network (such as shown at 112 in Fig. 1). Each of the first and second tasks has a status (such as, for example, "Not Started", "On track", "Complete", "In trouble", "At Risk", "On Hold" or "Cancelled") associated therewith. As shown in step S34, a first dependency relationship (such as, for example, SS, SF, FS or FF) is defined between the first and the second tasks (and between any of the other defined tasks). This defined first dependency relationship may then be stored in the database 110, as called for by step S44. The application embodying the present invention may then enable the remote updating of the status of the first and/or second tasks over the computer network 112 in the manner detailed above, as shown at S45. Not only may project participants remotely

update the status of the first and second tasks over the computer network (assuming proper authentication of the user and assuming that the user has permission to access and/or update the status the first or second tasks), but the database 110 and the server 111 may be configured to enable the user to also access the database 110 and to remotely define an Issue,
5 a Change Request and/or a Change Order, as shown at S46. The defined Issue, Change Request and/or Change Order may then be stored in the database 110. According to the present invention, an Issue identifies a problem within the project whose resolution is to be tracked, the Change Request identifies one or more steps to be taken pending authorization to resolve Issue and the Change Order identifies authorized steps to resolve the Issue. The
10 problem identified by the Issue, according to the present invention, may be thought of as a problem that was previously unidentified at the time when the first and second tasks were defined. As called for by step S47, the database 110 and/or the server 111 then may enable the remote definition of one or more second dependency relationships between the defined Issue, Change Request or Change Order and the first and/or second tasks. The defined
15 second dependency relationship(s) may then be stored in the database 110.

According to the present invention, permissions may be defined that delineate access rights for the first task, the second task, the Issue, the Change Request and/or the Change Order. In this manner, security may be maintained at all levels of the project and only those having the proper permission are allowed access to selected tasks. For example, if a portion
20 of the project is outsourced to an outside vendor, the vendor may be given permission (and required) to access the project hierarchy, but only to those tasks, Issues, Change Requests and/or Change Orders that pertain to the outsourced portion of the project for which the vendor has responsibility (for example). To insure that the vendor maintains current the

status of the tasks, Issues, Change Requests and/or Change Orders that pertain to the outsourced portion of the project for which the vendor has responsibility, the vendor may be required (by contract, for example) to make regular updates to the status of these items by accessing the database 110 over the network 112. The permissions assigned may further

5 define rights to remotely change the status of the first task and/or the second task and/or to remotely change the first dependency relationship between the first and/or second task. The permissions assigned to project groups and/or individual project participants may be crafted in most any manner consistent with the security requirements of the project, as those of skill in this art may recognize.

10 A lag time may be incorporated in the first and/or second dependency relationships, the lag time defining a lag time between the start and/or finish of at least two of the defined tasks, the Issue, the Change Request and the Change Order. For example, the dependency relationship between a first task and a second task may be defined in such a manner as to specify that the second task must be finished within 2 days of the second task finishing. The

15 lag quantity "2 days", therefore, would be stored in the database 110 along with the defined dependency relationship (e.g., SS, SF, FS, FF, for example) between the first and second tasks.

As shown in Figs. 1 and 3, the Web-enabled application embodying the present invention may maintain a selectively and remotely accessible graphical representation

20 (rendered on browser software, for example) of at least the first task, the second task, the first dependency relationship(s), the defined Issue(s), Change Request(s), the Change Order(s) and/or the second dependency relationship(s). Such graphical representation is preferably selectively truncated, masked or otherwise customized, depending upon the permission of the

person requesting access thereto. To insure that the defined tasks, Issues, Change Requests and Change Orders are accurately tracked and executed in a timely fashion, an identity of one or more entities (project team, a project member, a subcontractor and a vendor, for example) allowed access to and/or having responsibility for each of the defined first and second tasks, the defined Issue, Change Request or the Change Order is preferably specified and stored in the database 110. Additionally, one or more documents may be associated with and stored along the defined first and second tasks, the Issue, the Change Request and the Change Order. Such documents may further define and detail the task, Issue, Change Request and/or Change Order to which the document is associated. For example, the document may be a detailed design specification of a task, service, subcomponent and/or assembly to be produced by the task, Issue, Change Request and/or Change Order.

Fig. 5 is a flowchart of a method for managing complex projects, according to another aspect of the present invention. Specifically, the steps of the method detailed in Fig. 5 are those that may be taken by a person accessing the database 110 across the network 112 for the purpose of updating the status of or inputting a new task or tasks, Issue(s), Change Request(s) and/or Change Order(s). As shown in step S51, the database 110 may be accessed over the computer network 112, such as the Internet. As shown in Step S52, after proper authentication, the user accessing the database 110 may update the status of any task or selected tasks within the project, depending upon his or her assigned permission. As shown in step S53, the user, upon updating the status of a task, be given the opportunity to define an Issue or Issues, Change Request(s) and/or Change Order(s) and cause same to be stored in the database 110. Alternatively, the user may define such Issue(s), Change Request(s) and/or Change Order independently of any contemporaneous update of the status of any task, Issue,

Change Request and/or Change Order. The user may then, as shown at S54, define a dependency between the new Issue, Change Request and/or Change Order and any other previously defined task, Issue, Change Request and/or Change Order.

HARDWARE DESCRIPTION

5 Figure 6 illustrates a block diagram of a computing device 600 with which an embodiment of the present invention may be implemented. Examples of such computing devices are shown at reference numeral 118 in Fig. 1. Computing device 600 includes a bus 601 or other communication mechanism for communicating information, and a processor 602 coupled with bus 601 for processing information. Computing device 600 further comprises a
10 random access memory (RAM) or other dynamic storage device 604 (referred to as main memory), coupled to bus 601 for storing information and instructions to be executed by processor 602. Main memory 604 also may be used for storing temporary variables or other intermediate information during execution of instructions by processor 602. Computing device 600 may also include a read only memory (ROM) and/or other static storage device 606
15 coupled to bus 601 for storing static information and instructions for processor 602. A data storage device 607, such as a magnetic disk or optical disk, may be coupled to bus 601 for storing information and instructions. A communication device 608, such as a modem or network (such as Ethernet, for example) card is also coupled to the bus 601 to provide access to a network, such as shown at 112 in Fig. 1.

20 The computing device 600 may also be coupled via bus 601 to a display device 621, such as a cathode ray tube (CRT), for displaying information to a computer user. An alphanumeric input device 622, including alphanumeric and other keys, is typically coupled to bus 601 for communicating information and command selections to processor 602. Another

type of user input device may be the user's own voice or cursor control 623, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 602 and for controlling cursor movement on display 621.

The present invention is related to the use of computing device 600 to manage
5 complex projects. According to one embodiment, the processing may be carried out by one
or more computing devices 600 in response to processor(s) 602 executing sequences of
instructions contained in memory 604. Such instructions may be read into memory 604 from
another computer-readable medium, such as data storage device 607 and/or from a remotely
located server. Execution of the sequences of instructions contained in memory 604 causes
10 processor(s) 602 to implement the functionality described above. In alternative
embodiments, hard-wired circuitry may be used in place of or in combination with software
instructions to implement the present invention. Thus, the present invention is not limited to
any specific combination of hardware circuitry and software.

While the foregoing detailed description has described preferred embodiments of the
15 present invention, it is to be understood that the above description is illustrative only and not
limiting of the disclosed invention. Those of skill in this art will recognize other alternative
embodiments and all such embodiments are deemed to fall within the scope of the present
invention. Thus, the present invention should be limited only by the claims as set forth
below.